ES Project

Submitted to:

prof. Bassem Ibrahim

Submitted by:

Mohamed Akram AbdelFattah 211

Ahmed Emad 108

Ahmed Sayed Seif 106

Ahmed Hamdy Nossir 104

Ayman Mohamed Reda 120

Gaser Ashraf 124

Abdullah Mahmoud 142

Omar AbdElFattah 204

Mariem Naser 223

Mariem Mohamed Zein 222

Youssef Atef Shabrawy 242

2022

Hardware Documentation

Components :

- Test motors (weak motors) x2
- 12v - 12000rpm brushed motors (rs-380) x2
- Arduino Uno and its connector

- Plastic d6 wheels (toy wheels) x2
- Gearbox with compatible shaft x2
- IR sensors (cny-70) x5
- H-bridge x1
- 3.74v batteries x3
- Battery holder x1
- Potentiometer x1
- Push button x1
- Button switch x1
- PCB x1
- Breadboard x1
- Chassis
- Different kinds of wires
- Photo-interrupt x2
- Encoder gears x2
- Others (Avo-meter, Soldering kit, LEDs, 3D printed stuff, charger, electric drill/saw, sticky chemicals, etc.)

Budget	1200 EGP
Expenses	1265
2 rs-380 sh motor	200
2 Gear box motors	50
2 Encoders	30
2 Encoder gears	60
3 Batteries	90
Battery holder	15
Charger	50
Motor driver	65
10 IR sensor	120
2 PCB	30
Breadboard	30
2 wheels	30
Wires	150
3D printing	210
Resistors	15
Flux	10
2 Gear box motors	70
Leds	40
Glues (double face, uhu, epoxy, electrical tape, etc.)	~70

Motors:

We initially used the small motors to ensure our algorithm is working correctly, however, they can't really be reliable when our weight exceeds a certain point.
So we used strong motors and fixed them into the gearbox that came with the weak motors to drive very strong torque. No motors were

compatible with the gearboxes, neither are the wheels so we literally HACKED it.

Sensors:

We were faced with a huge variety of sensors, sensor array, cny-70, normal transmitter/detector module, just the detector, etc.
We went for the cny-70 due to the following perks:

- We can control spacing in between sensors.
- Both the emitter and the detector antenna are embedded inside the plastic, not fully exposed like bulbs.
- It had walls around to prevent outer sources of IR from reaching the detector, also it had protection against day-light.

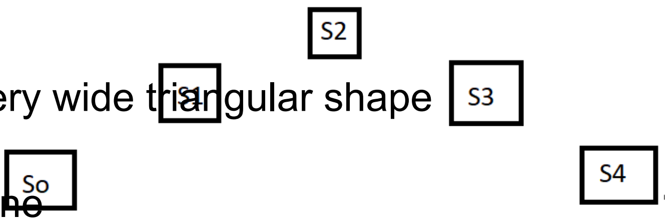
Though, it wasn't very user friendly as it had no labels and any misconnection can cause the sensor to just die.
Another downside (for all sensors not only cny-70) is that they must be away from the surface by 1cm max.

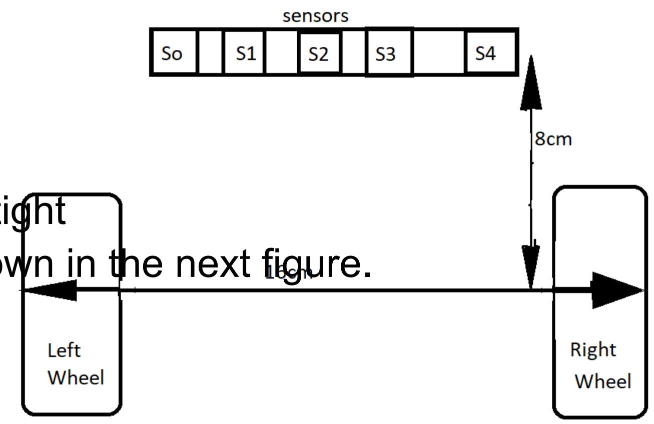
Why did we use 5 sensors ?

- For LF, the more sensors there are, the more resolution you can get from any curve, the more accurately you can determine the error and hence determine the correct speed ratio between both motors. We will use 7 sensors in future work.
- For MS, the 3 middle sensors are just for error correction when moving in a straight line, but both outsider sensors are for detecting the sharp 90° lines.

How did we place the sensors ?

- For LF, we initially placed them in a very wide triangular shape as in the figure.
  Then we decided to go for a straight line
  Of sensors, to get better accuracy
  Specifically For the competition.





(as there were no sharp turns or tight curves) the final placement is shown in the next figure.

- For MS, we placed the outer 2 sensors at exactly half The distance between wheels, So when we take a sharp 90° turn, We just stop a motor and operate the other one and our robot will always land straight and centered on the new line (imagine turning in a circle of radius 8cm).

  We used a PCB only for sensor connections so as to achieve robustness and be 100% sure there will be zero short circuits and to ease up our reassembly process, we don't have to get deep into sensors' connection again…

  -ps: soldering the PCB was a nightmare-

We also 3D printed a robust and tidy holder for our sensors and pcb with flexible dimension options (proximity with the ground, changing angles, changing the sensor distances from each other, changing the position of all sensors at once) but we didn't have time to put it in our project as it was finished a day ago.

Microcontroller:

We used Arduino Uno for its convenience and ease of use, Although it has some limitations listed below:

- Very small memory (1 KB)

- Only 5 analog input ports
- Somewhat slow (other microcontrollers have higher frequencies)
- Containing only one ADC (we must put delays between analog inputs)
- Large size
- Pins are inconvenient for final design (we want to solder our wires to avoid excessive movement)

For future work, we will use the stm32 blue pill as it is cheaper, smaller and with greater capabilities.

Potentiometer:

We used a simple potentiometer to control the threshold that will be applied to sensors' readings.

Push Button:

- For LF, we use it to just operate the robot.
- For MS, we use it to operate the robot and to switch from exploring the maze to processing the shortest path and traversing it.

Button Switch:

- Just to turn on/off our motor driver to avoid getting the robot to randomly move while reprogramming our arduino.

Power Source:

- We used 12v lithium batteries (3 batteries 3.74v) to operate the strong motors, and a regulator to power our arduino

(12v → 5v), we cannot always connect the arduino to a laptop.
- The battery holder can take up to 3 batteries and can be simply connected to an external charger.
- Our batteries cannot drive solid 12v but they can operate our motors.
- Our batteries had a short lifetime as we were always overcharging them.

Wires:

- We used normal copper wires, m-m, m-f, f-f, crocodile and soldering wires.

Breadboard & PCB:

- We used a single breadboard for our IO like potentiometers, buttons, switches, etc.
- Our PCB is only for sensors' connections and it is hidden under our chassis.

H-bridge:

To control the speed and direction of our motors we have decided to use L298N H-Bridge and not to use the Arduino shield (L2983) for many reasons:

- ★ H-Bridge is very flexible with outside voltages (it can work with 9 or 6 volts not only exact 12 volt).
- ★ H-Bridge has a maximum current of 2A while Arduino-shield only 600 mA.
- ★ Arduino will be hidden in the case of the Arduino shield so it will be harder to deal with.
- ★ H-Bridge is simpler in code and connections.

However, H-Bridge has some problems:

- ★ It is relatively big in size.
- ★ In the case of using more powerful motors it won’t support 24v so we will have to use a relay which is complex and will need more control.

Chassis:

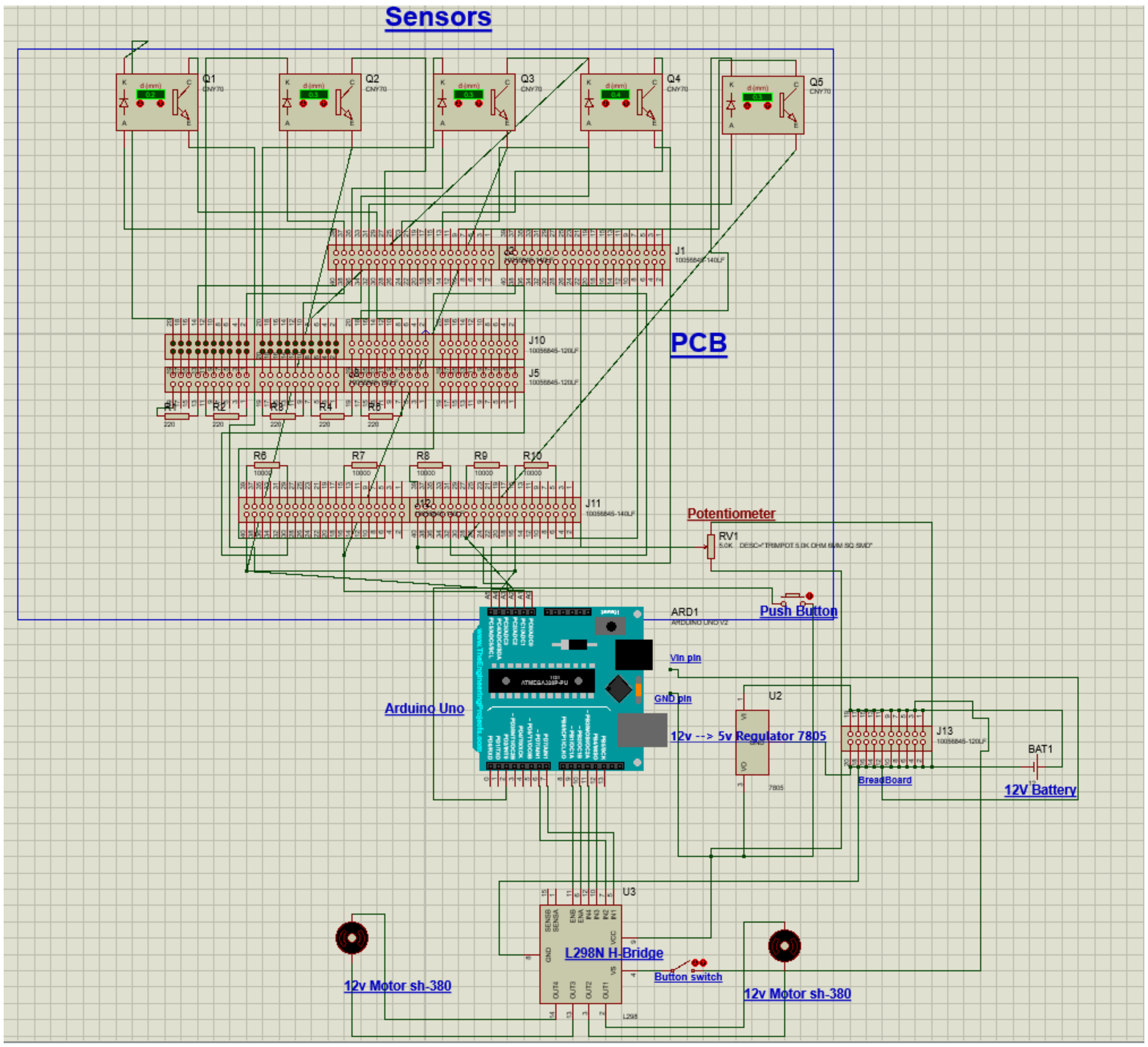
It is 24cm x 20cm x 10cm. At first it was larger in size and this was to cause problems while turning right or left, especially in the maze. So, it was reduced to a smaller size. Yet, it can be further cut to a smaller size. Which would result in better control.

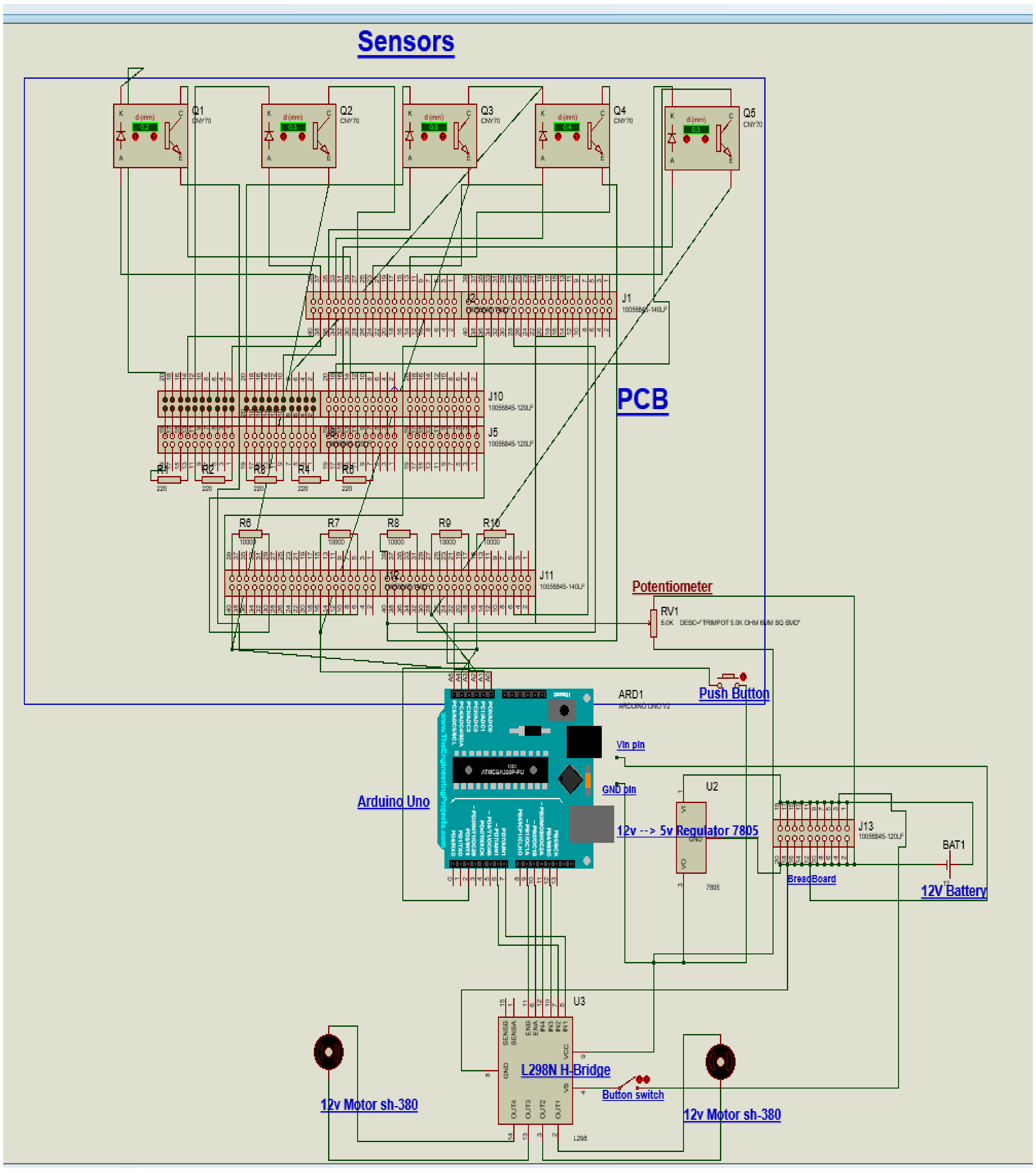
An omni wheel was used in its front side to make turning right and left easier. Motors are bolted into the back of it. Each motor is around 12000 rpm. But they have low torque. So, to overcome this, 3D printed motor holders and gearboxes were used to make these motors fixed to the chassis. And motors were attached to the gearboxes to increase their torque.

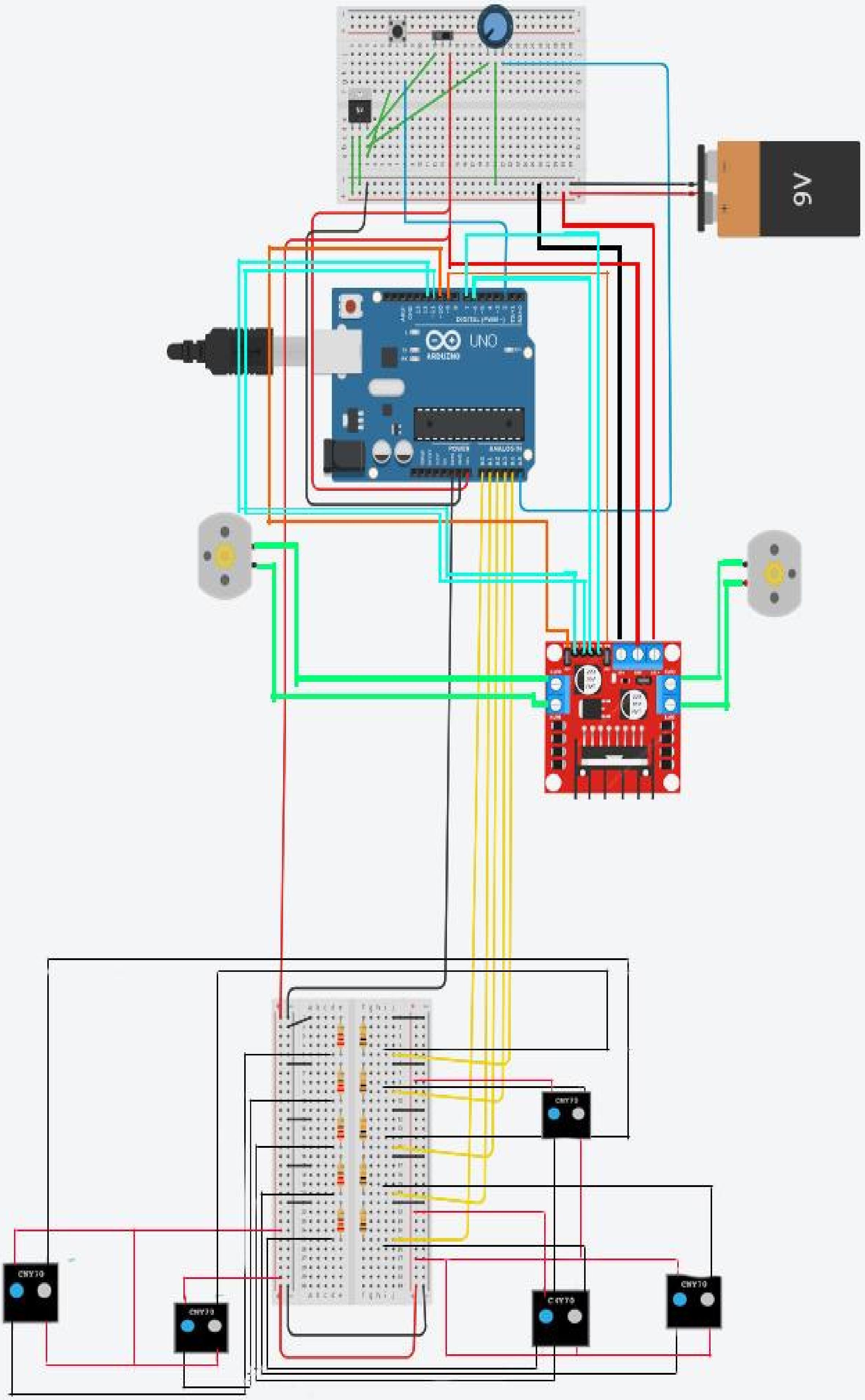
On top of the chassis, the arduino chip, UI breadboard, H-bridge and batteries all are put. While on bottom, the PCB, sensors, omni-wheel, and motors are bolted.

Distance between both wheels is 16cm although our wooden chassis is wider to provide protection.

Schematic:


Sensors
Q1
CNY70
Q2
CNY70
Q3
CNY70
Q4
CNY70
Q5
CNY70
J1
J2
J10
J5
J11
J12
J13
PCB
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
Potentiometer
RV1
Push Button
ARD1
ARDUINO UNO V2
Vin pin
GND pin
Arduino Uno
U2
12v --> 5v Regulator 7805
BreadBoard
BAT1
12V Battery
U3
L298N H-Bridge
Button switch
12v Motor sh-380
12v Motor sh-380


Sensors
Q1
CNY70
Q2
CNY70
Q3
CNY70
Q4
CNY70
Q5
CNY70
J1
10056845-140LF
J2
J10
10056845-120LF
J5
10056845-120LF
PCB
R1
R2
R3
R4
R5
220
R6
R7
R8
R9
R10
10000
J11
10056845-140LF
J12
Potentiometer
RV1
5.0K
Push Button
ARD1
ARDUINO UNO V2
Arduino Uno
Vin pin
GND pin
U2
12v --> 5v Regulator 7805
7805
J13
10056845-120LF
BreadBoard
BAT1
12V Battery
U3
L298N H-Bridge
L298
Button switch
12v Motor sh-380
12v Motor sh-380



9V
UNO
ARDUINO
CNY70
CNY70
CNY70
CNY70
CNY70

Software Documentation

Line Follower:

The code flow, First we were reading the values from the sensors then thresholding these values (0->white , 1->black) that how we can differentiate the black line and the white ground. Then we pass the sensors reading to the PID function that decides the error in the movements (0-> exactly on the line ,1 to 4 the car is moving towards right by different angles , -1 to -4 the car is moving towards left by different angles). This error is being used to decide the speed to be added or subtracted to each motor from the initial speed. If the car is going right then we need to make the left motor’s speed higher than the right one so that it can move correctly. Last thing was to decide the speed of each motor. This is done by passing the speed to be added or subtracted from each motor to decideSpeed function. If the error is 0 the car will continue moving straight. If the error was -1 or -2 the speed of the right motor will be higher than the speed of the left motor. If the error was 1 or 2 the speed of the left motor will be higher than the speed of the right motor. If it was -3,-4,3 or 4 it would be the same but with much higher speed differences.

Maze Solving:

The code flow, First we were reading the values from the sensors then thresholding these values (0->white , 1->black) that how we can differentiate the black line and the white ground. Same as in line following also we used the PID to put the car on the track when it moves slightly to left or right. The algorithm we used is following LHR The car is moving straight until it finds an intersection or dead end. When one of them happens the sensors reading tells us which direction we should follow. There is a priority in deciding the direction of the next move and it is left, straight, right and back. For the car to turn left the left motor stops and the right motor moves. And the opposite to turn right. Going back is done by moving one motor in the opposite direction of the other one. We store the path of the car to make simplifications on it after we solve the maze. After we solve the maze we first simplify the path the car took. LBL = S ,LBR = B ,LBS = R, RBL = B, SBL= R and SBS = B. If the car took the path Left, Back then Left it should have gone straight from the beginning. This simplification is done many times until we reach the most simplified path. Lastly, we make the car run one more time with the simplified path.

Team Contributions :

- Mohamed Akram AbdElFattah:

  - Hardware specialist
  - Sensor placement idea
  - Calculating correct dimensions for sensors
  - Sensor assembly
  - General assembly
  - Component choice
  - Contributed in Documentation

- Marim Naser:

  - Gearbox gear ratio research
  - Sensor connection
  - PCB inspection
  - Choice for chassis shape & Holder’s design
  - General assembly
  - Contributed in Documentation

- Ayman Mohamed Reda Mohamed:

  - Teamspeak
  - PCB Soldering
  - Wiring & General assembly
  - Motor and its compatible Gearbox research
  - 3D designs

- Hardware testing
- End-to-end testing
- Hardware modifications and updates
- Proteus simulation
- Contributed in Documentation

- Mariem Mohamed Zein:
    - Motor and its compatible Gearbox research
    - Sensor connections and debugging
    - Contributed in Soldering
    - Code for testing Sensors
    - Contributed in Documentation
- Ahmed Sayed Seif: Software Specialist
    - Handled most of the LF code
    - Ensuring clean code and functionalizing the entire code
    - Tuning parameters
    - Documentation of the LF and MS
    - Commenting and making both codes Readable
- Ahmed Emad ElDin:
    - Final touches in LF code
    - Contributed in MS code
    - Documentation of the LF and MS
    - Commenting and making both codes Readable
    - Testing the MS code along with Ayman

- Gaser Ashraf Atress:
  - Contributed in MS code
  - Responsible for the clean 90° turns (left and right)
  - Contributed in Sensor test code

- Youssef Atef ElShabrawy:
  - Encoder test code and wiring
  - 3D-printing needed designs.
  - Buying essential components
  - Hardware testing and debugging
  - General assembly

- Omar Abd-ElFattah:
  - Buying essential components
  - Ideas contributions
  - Comparing prices & quality and our requirements

- Youssef Mostafa ElMahdy:
  - Select suitable motors and sensors
  - Select arduino, batteries, holder and responsible for their assembly

- Encoder test code and wiring
- General assembly
- Hardware testing and debugging
- Contributed in sensor Code
- Contributed in Documentation

- Abdullah Mahmoud:
  - Contributed in MS code
  - Responsible for moving forward and correcting the car position
  - Contributed in Sensor test code

- Ahmed Hamdy Mohamed Nossir :
  - Contributed in MS code
  - Responsible for turning back selecting the direction
  - Contributed in Sensor test code